REMARKS

Applicants have amended their claims to more particularly point out their invention, and request the Examiner to reconsider and allow their patent application in view of the amendments and the following remarks.

As an initial matter, applicants attach a further set of formal drawings that corrects transcription errors in the previously-filed formal drawings.

Regarding the Rejections Under 35 USC §112 & 35 USC §101

The Examiner has objected to applicants' use of the term "interlaced" because, the Examiner alleges, applicants have used it in a manner that is "repugnant to the usual meaning of that term." See Office Action at 2-3. Applicants' specification states that in a preferred exemplary embodiment, a variable called "INTERLACE" is defined to be a positive constant integer such as 2, 3 or 4 for example. See page 15. However, that "definition" of a particular variable does not define what applicants mean by the term "interlace" in their title and in certain of their claims (e.g., claims 9 and 10). As one of ordinary skill in the art would understand by reading additional portions of applicants' specification (e.g., see page 16 and Figure 6), what applicants disclose is the concept of interpolating a subset of all possible texels each frame (e.g., so that one third of the texels are interpolated during a first frame time, another third of the texels are interpolated during a second frame time, and the remaining third of the texels are interpolated during a

third frame time). See for example page 16 lines 18-25. This usage of "interlace" is not repugnant to its standard meaning.

Applicants do not understand the Examiner's rejection under 35 USC §112, first paragraph. Applicants respectfully point out that their specification does adequately disclose a well-established utility -- i.e., generation of interactive computer-generated animation displays for use in video games and the like. See page 2, lines 15-17 for example. Such a use certainly provides an accepted and well-established utility under the standards of 35 USC §101. Moreover, applicants' disclosure provides detailed flowcharts and other description that would clearly have enabled someone of ordinary skill in the art to make and use the invention. See for example Figures 3 and 7 and accompanying text. Applicants request the Examiner to reconsider and withdraw this rejection.

Applicants respectfully disagree with the Examiner's assertion that their claims do not "set forth any steps involved in the method/process….." Office Action at 3. Applicants' claims specifically recite a number of steps or structures (as the case may be). To more particularly point out the invention, applicants have added a displaying step (or structure) to each of their claims except for claim 34, which claim reads on a data storage device as recommended by MPEP §2106 at 2100-13.

Applicants respectfully submit that their amended claims provide processes that are limited to a practical application in the technological arts and are therefore statutory. See MPEP §2106 at 2100-17 and following. Should the Examiner wish to persist in this

rejection, applicants respectfully request an interview with the Examiner and the Examiner's supervisor to discuss these issues and hopefully avoid the need to burden the written record with unnecessary additional written argument.

In response to the Examiner's objections to claims 1-13 as being directed to a "method" as opposed to a "process", applicants have made the change the Examiner suggests. By doing so, applicants specifically intend to avoid invoking 35 USC §112, sixth paragraph. Applicants have also corrected claim 28 as suggested by the Examiner, and have amended their specification to correct some minor typographical errors.

Obviousness Rejection

Applicants have amended each of their independent claims 1,14, 27, and 32 to further recite, among other things, that an incremental interpolation factor is pre-computed based on a texture map that has been pre-decomposed into its respective color components. As described in applicants' specification at page 2, line 2, conventional texture morphing tends to be computationally expensive -- which has in the past effectively prevented resource-constrained real-time graphics systems such as home video games systems and personal computer graphics cards from providing real-time texture morphing functionality. Applicants have solved this problem by, for example, providing off-line texel component decomposition and incremental interpolation. See page 2, lines 10 and following. These techniques provide texture morphing procedures

that are fast enough for real-time computer animation and simulation even in resource-constrained home video game systems such as shown in Figure 1.

The Iourcha et al. reference does not teach or suggest the claims as amended. Initially, notwithstanding Iourcha et al's use of the term "morphing" at column 1, line 52, what Iourcha et al. really discloses is a tri-linear texture filtering technique for generating views of the <u>same</u> object at <u>different levels of detail</u>. See column 1, line 39 and following. Furthermore, applicants can find no indication in Iourcha et al. that any texture map is decomposed in advance into components and that any incremental interpolation values are pre-computed. In applicants' preferred embodiment, interpolation parameters are generated beforehand so that the calculations that need to be performed during run time are relatively simple (e.g., in the exemplary embodiment, by simply adding an incremental integer value to a pre-decomposed texel component to achieve an interpolated value for the next display frame). In contrast, the Iourcha et al. tri-linear texture filtering technique relies on fairly complex equations to dynamically interpolate between two levels of detail at run time. See columns 8, 9, 14 and 15, for example, Iourcha et al. therefore does not appear to help solve the problem that applicants have solved, and in any event does not teach or suggest applicants' claimed invention.

The Hoppe reference is directed to "geomorphing" by using view-dependent progressive-meshes. See column 1, line 35 and following. Hoppe does not disclose texture morphing as claimed herein. Rather, as applicants understand the Hoppe

683549

reference, it is the meshes -- not textures --(e.g., network or fabric of triangles) that are "morphed." Conventional texture mapping is then used to map textures onto the "morphed" meshes. This technique does not teach or suggest applicants' claimed invention. In particular, applicants do not understand the Examiner's assertion that "[i]t would have been obvious to one of ordinary skill in the art at the time of the invention of Hoppe to include texel components in the scalar attributes because texture values are attributes of a pixel." See Office Action at 8. Hoppe appears to be interpolating between parameters relating to polygon mesh vertices -- not texture morphing as claimed herein.

Conclusion

All outstanding issues have been addressed and this application is believed to be in condition for allowance. Should any minor issues remain outstanding, the Examiner should contact the undersigned at the telephone number listed below so they can be resolved expeditiously without need of a further written action.

Respectfully submitted,

NIXON & VANDERHYE P.C.

By: ______________________

Robert W. Faris
Reg. No. 31,352

RWF:ejs
1100 North Glebe Road, 8th Floor
Arlington, VA 22201-4714
Telephone: (703) 816-4000
Facsimile: (703) 816-4100

VERSION WITH MARKINGS TO SHOW CHANGES MADE

IN THE SPECIFICATION

Page 1, after the title, "INCREMENTAL INTERLACE INTERPOLATION FOR TEXTURE MORPHING," insert the following:

CROSS-REFERENCES TO RELATED APPLICATIONS

Priority is claimed from application no. 60/169,309 filed 1/28/2000, which is incorporated herein by reference.

The paragraph beginning on page 2 line 18:

In accordance with one aspect of an exemplary embodiment of the present invention, incremental interpolation techniques are used to reduce repetitive and heavy floating point number calculations/conversions associated with the typical texel blending/morphing process. The preferred embodiment computes an incremental morph parameter t for each texel component based on a previous value(s) and change rate (e.g., image frame rate and the time duration of the morphing process). Initial and incremental morph parameter values can be computed in advance for each texel component during a preliminary morph preparation background process. Then, during a subsequent real-time morphing process, these initial and incremental parameter values are applied incrementally to morph the texel components toward target texel component values.

The paragraph beginning on page 3 line 17:

In accordance with another aspect of the exemplary preferred embodiment of the invention, incremental interpolation can be applied each frame time or other morphing period to less than all of texels being morphed. For example, some texels can be incrementally interpolated during a particular frame, other texels can be incrementally interpolated during a subsequent frame, etc . -- so that all texels are incrementally interpolated across a certain (preferably small) number of frames without requiring each texel to be interpolated every frame. Such interlacing of incremental interpolation can significantly reduce computational load without introducing significant image artifacts.

The paragraph beginning on page 4 line 9:

One significant and advantageous application of the present invention is to allow dynamic generation of a virtually infinite number of video game characters and other textured objects "on the fly" using morphing procedures within home video game systems, personal computer graphics cards, and other inexpensive graphics systems. It is possible to pre-construct a number of objects with certain geometry and textures as primary sources and targets (" morph terminals"), and then use the texture morphing features ~~provided by this invention~~ to smoothly transform textures in real time between such objects to generate a sequence of continuous intermediate objects along morphing paths between the morph terminals. Since the preferred embodiment stores each set of intermediate morphed texture values as a texture map, any such intermediate texture map can be used as a source texture map for a further texture morphing operation along the same or different morph path.

The title on page 8 line 18:

A New Texture Morphing Process Provided by the Preferred Embodiment of the Present Invention

The paragraph beginning on page 8 line 19:

Figure 3 is a flowchart of an overall process provided by a presently preferred example process 100 of a preferred embodiment of the invention. Process 100 is divided into two overall stages: an authoring stage 102, and a run-time stage 104.

IN THE CLAIMS

1. (Amended) A ~~process~~method for morphing and displaying a texture comprising:

pre-decomposing at least some texels of a texture map into respective texel color components;

predetermining, based on said decomposed texture map and target texel color component states defined by a target morph texture map defining a target morph texture, ~~an~~ at least one incremental morph parameter corresponding to ~~a~~ said respective texel color components, ~~and~~

~~in response to~~ using said incremental morph parameter~~,~~ during real-time imaging to incrementally interpolate~~ing~~ said texel color components toward ~~a~~ target texel color component states through at least one intermediate morph texel color component state; and~~.~~

683549

displaying an image based at least in part on said intermediate morph texel color state.

2. (Amended) A ~~process~~method as in claim 1 wherein said incrementally interpolating ~~step~~ comprises repetitively adding said predetermined incremental morph parameter to said predetermined texel components to produce a corresponding sequence of intermediate morph texel component states.

3. A ~~process~~method as in claim 1 wherein said incrementally interpolating ~~step~~ comprises using an integer arithmetic calculation to repetitively increment or decrement said plural texel components based on said predetermined incremental morph parameter.

4. (Amended) A ~~process~~method as in claim 1 wherein said predetermining ~~step~~ calculates said incremental morph parameter as the amount of change in said texel components for each successive time period within a morphing procedure, and said incrementally interpolating ~~step~~ changes said texel components in response to the integer portions of said incremental morph parameters.

5. (Amended) A ~~process~~method as in claim 4 wherein said successive time periods comprise image frame times.

6. (Amended) A ~~process~~method as in claim 4 wherein said incrementally interpolating ~~step~~ conditions said change in said texel components based on which of said

successive time periods has occurred within said morphing procedure to ~~reduce~~ minimize the number of calculations required to morph said texture~~d surface~~.

7. (Amended) A ~~process~~method as in claim 4 further including selectively adding ~~1~~ integers to or subtracting ~~1~~ integers from said integer portions to reduce approximation errors in the context of integer arithmetic operations.

8. (Amended) A ~~process~~method as in claim 1 wherein said incremental interpolation ~~step~~ comprises incrementing or decrementing said texel components by integer approximations of said determined morph parameters, and compensating for approximation errors by performing at least one floating point operation to set said texel components to~~ a~~ target texel component values.

9. (Amended) A ~~process~~method as in claim 1 wherein said incremental interpolation ~~step~~ selectively interpolates said texel components based on an interlace factor.

10. (Amended) A ~~process~~method as in claim 1 further including conditioning said incremental interpolation step based on an interlace factor.

11. (Amended) A ~~process~~method as in claim 1 further including calculating a frame counter corresponding to said texel components, and selectively incrementing or decrementing said texel components in response to said frame counter.

12. (Amended) A ~~process~~method as in claim 1 further including the preliminary step of storing said decomposed texel components in~~-a~~ separate texel component arrays.

13. (Amended) A ~~process~~method as in claim 12 wherein said texel components comprises ~~one of a~~ red, green ~~or~~ and blue color values and an alpha value.

14. (Amended) A system for morphing and displaying a texture~~d surface~~ comprising:

a color decomposer that pre-decomposes at least some texels of a texture map into respective texel color components;

a predeterminer that predetermines ~~-an-~~incremental texture component morph parameters based on said decomposed texels and target morph texture texel color component states,~~corresponding to a texel component, and~~

an incremental interpolator that incrementally interpolates, in response to said predetermined incremental morph parameters, said texel components toward ~~a~~ said target texel color component states through at least one intermediate morph texel component state~~.~~; and

a real-time image generator that generates a display based at least in part on said intermediate morph texel component state.

15. (Amended) A system as in claim 14 wherein said incremental interpolator repetitively adds said incremental morph parameters to the texel components to produce a corresponding sequence of intermediate morph texel component states.

16. A system as in claim 14 wherein said incremental interpolator comprises an arithmetic calculator that performs a repetitive integer arithmetic calculation to repetitively increment or decrement said plural texel components based on said determined incremental morph parameters.

17. (Amended) A system as in claim 14 wherein said incremental interpolator calculates said incremental morph parameter as the amount of change in said texel components for each successive time period within a morphing procedure, and changes said texel components in response to the integer portion of said incremental morph parameters.

18. (Unamended) A system as in claim 17 wherein said successive time periods comprise image frame times.

19. (Amended) A system as in claim 17 wherein said incremental interpolator conditions said change in said texel components based on which of said successive time periods has occurred within said morphing procedure to as to reduce the number of calculations required to morph said texture~~d surface~~.

20. (Amended) A system as in claim 17 further including an adder that selectively adds or subtracts 1 relative to said integer portion to ~~reduce~~ minimize approximation errors in the context of integer arithmetic operations.

21. (Amended) A system as in claim 14 wherein said incremental interpolator increments or decrements said texel components by integer approximations of said determined morph parameters, and compensates for approximation errors by performing at least one floating point operation to set said texel components to a target texel component value.

22. (Amended) A system as in claim 14 wherein said incremental interpolator selectively interpolates said texel components based on an interlace factor.

23. (Amended) A system as in claim 14 further including a conditioner that conditions said incremental interpolation ~~step~~ based on an interlace factor.

24. (Unamended) A system as in claim 14 further including a frame counter corresponding to said texel component, and wherein said incremental interpolator selectively increments or decrements said texel component in response to said frame counter.

25. (Unamended) A system as in claim 14 further including a separate array storing said texel component arrays.

26. (Amended) A system as in claim 25 wherein said texel components comprise~~s~~ ~~one of a~~ red, green or blue color values and an alpha value.

27. An efficient texture~~3D~~ morphing ~~process~~method for morphing ~~3D~~ and displaying textures~~d objects displayed~~ using a real-~~-~~time interactive 3D graphics system including user-manipulable controls, said system displaying at least one 3D texture-mapped object based at least in part on a morphed texture map comprising plural texels ~~each comprising plural texel components~~, said ~~3D~~ texture morphing ~~process~~method including:

(a) before imaging time, pre-decomposing said texture map into plural texel components and precalculating incremental morph parameter values for the texel~~s~~ components; ~~of said texture map; and~~

(b) ~~in~~ during real-~~-~~time imaging, incrementally changing the values of said plural texel components over time based on said calculated incremental morph parameter values~~.~~; and

(c) during real-time imaging, generating images in real time based at least in part on said incrementally-changing texel component values.

28. (Amended) A ~~process~~method as in claim 27 wherein said calculating step (a) comprises calculating the value of $\Delta r = (TC-SC) / (FR * T)$, where SC is the source texel

component value, ~~S~~TC is the target texel component value, FR is the frame rate and T is the morphing duration.

29. (Amended) A ~~process~~method as in claim 28 wherein said incrementally changing ~~step~~ comprises repetitively incrementing or decrementing said plural texel component values by uniform amounts at a first predetermined frequency based on the integer portion of Δr, and adding or subtracting a further integer value at a further predetermined frequency less than said first predetermined frequency.

30. (Amended) A ~~process~~method as in claim 29 wherein said first and second predetermined frequencies are each based on image frame rate.

31. (Amended) A ~~process~~method as in claim 29 wherein said second predetermined frequency is based on a frame counter that counts a predetermined number of image frames.

32. (Amended) An efficient ~~3D~~ texture morphing ~~process~~method for morphing and displaying ~~3D textured objects displayed~~textures using a real time interactive 3D graphics system including user-manipulable controls, said system displaying at least one 3D object based at least in part on a morphed texture map comprising plural texels each comprising plural texel components, said ~~3D~~ texture morphing ~~process~~and display method including:

(a) before real-time imaging, pre-decomposing said texels into plural texel components and precalculating incremental morph parameter values for said texel components, including rounding down calculated incremental interpolation values to the closest integer values to provide integer results and calculating period counter values based on non-integer remainders of said calculated incremental interpolation values;

(b) at least in partial response to user interaction with said controls, changing texel component values at a first periodic frequency based on said integer results; ~~and~~

(c) at least in partial response to said period counter, further changing said texel component values at a second periodic frequency less than said first periodic frequency to compensate for approximation errors introduced by step (b)~~.~~; and

generating an image display based at least in part on said changed and further changed texel component values.

33. In a real-time interactive graphics system including at least one user-manipulable control, a ~~process~~method for generating animation objects in real time by morphing a source texture map including plural texels each having plural components, into a target texture map including plural texels each having plural components, said ~~process~~method comprising:

(a) calculating incremental morph parameter values for texels of said first texture map, and incrementally interpolating the value of said plural texel components of said

first texture map over time based on said calculated uniform incremental morph parameter values so as to morph said first texture map toward said second texture map;

(b) using an intermediate texture map generated by step (a) to texture map an animation object; ~~and~~

(c) controlling at least one of the displayed orientation and position of said texture-mapped animation object at least in part in response to user manipulation of said control~~.~~; and

(d) generating an image based at least in part on said controlled texture-mapped animation object.

34. (Amended) A storage device for use with a real-time interactive graphics system including at least one user-manipulable control, said storage device storing information used by said system for generating animation objects in real time by morphing a source texture map including plural texels each having plural components, into a target texture map including plural texels each having plural components, said storage device comprising:

a first storage portion that stores information controlling said system to calculate incremental morph parameter values for texels of said ~~first~~ source texture map, and to incrementally interpolate the values of said plural texel components of said first texture map over time into values of plural texel components of said target texture map by uniform integer amounts based on said calculated incremental morph parameter values so

as to morph said ~~first~~ source texture map through at least one intermediate texture map toward said ~~second~~ target texture map;

a second storage portion that stores information controlling said system to use said intermediate texture map to texture map an animation object; and

a third storage portion that stores information controlling at least one of the displayed orientation and position for display of said texture-mapped animation object at least in part in response to user manipulation of said control.